ELECTRIC POWER RESEARCH INSTITUTE

IEC 61850 Protocol Driver Agents

Open Source Software to Support Testing and Development of IEC-61850 in Smart Inverters

3002013625

IEC 61850 Protocol Driver Agents

Open Source Software to Support Testing and Development of IEC-61850 in Smart Inverters

3002013625

Technical Update, December 2018

EPRI Project Manager

B. Ealey

ELECTRIC POWER RESEARCH INSTITUTE
3420 Hillview Avenue, Palo Alto, California 94304-1338 ▪ PO Box 10412, Palo Alto, California 94303-0813 ▪ USA
800.313.3774 ▪ 650.855.2121 ▪ askepri@epri.com ▪ www.epri.com

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

THE FOLLOWING ORGANIZATION, UNDER CONTRACT TO EPRI, PREPARED THIS REPORT:

Electricité de France R&D

This is an EPRI Technical Update report. A Technical Update report is intended as an informal report of continuing research, a meeting, or a topical study. It is not a final EPRI technical report.

NOTE

For further information about EPRI, call the EPRI Customer Assistance Center at 800.313.3774 or e-mail askepri@epri.com.

ACKNOWLEDGMENTS

The following organization, under contract to the Electric Power Research Institute (EPRI), prepared this report:

Electricité de France R&D
EDF Lab Paris – Saclay
7 Bd Gaspard Monge
91120 Palaiseau, France

Principal Investigator
Q. Morel
K. Madoret

This report describes research sponsored by EPRI.

This publication is a corporate document that should be cited in the literature in the following manner:

IEC 61850 Protocol Driver Agents: Open Source Software to Support Testing and Development of IEC-61850 in Smart Inverters. EPRI, Palo Alto, CA: 2018. 3002013625.

ABSTRACT

As grid codes and utility programs are increasingly requiring end-use devices and their control systems to use open standards it is more and more important that validation tools, simulation tools, and reference implementations are available to foster growth in the industry. Experience has shown that uptake can be slow without requirements to use a specific standard. For adoption to succeed a three-faced approach can be used.

1. Continuous technology transfer will help inform the industry of standards and teach them about the benefits of applying them.
2. Creation of open source code that vendors/manufacturers can implement in their products will likely accelerate adoption of the protocol by minimizing the time required for products to appear on the market.
3. Certification frameworks are key to ensuring protocols meet the standards.

In this project Electricité de France R&D and the Electric Power Research Institute explore the first two. The objective of this project is to develop open source code to implement IEC-61850 in both servers and clients and perform the necessary technology transfer to bring awareness of these resources. This report provides an overview of open source software to support testing and development of IEC-61850 in smart inverters. The software is still in development. This report serves as a guide for those involved in beta-testing of the software.

Keywords
Smart Inverters
Interoperability
IEC-61850
Communications
Open Source

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1
INTRODUCTION

As grid codes and utility programs are increasingly requiring end-use devices and their control systems to use open standards[1] it is increasingly important that validation tools, simulation tools, and reference implementations are available to foster growth in the industry. Utilities need tools that can be used to evaluate these products and their capabilities to ensure they meet the requirements of RFPs, interconnection agreements, and intended use cases.

Many products claim to fulfil specific communication requirements, but unless there are independent tools to evaluate these claims, it is unlikely that multiple brands or types of equipment will interoperate. Third parties have created test tools however not all protocols have test tools or established certification frameworks. Even when protocols are implemented properly there are other, non-standardized practices that can lead to barriers including custom control algorithms or other proprietary management techniques. Test tools help identify these barriers prior to deployment in the field.

The integration of distributed energy resources and bi-directional demand response technologies into deployments with other utility control system using open communication standards are new and have few case studies. Utilities, national labs, and industry researchers are exploring and validating these new use cases through laboratory and field testing. EPRI's reference control systems and device simulators make testing of these use cases simple and enable advanced hardware and software in the loop testing. These test tools have been deployed in National Labs including Sandia and NREL and utilities including SMUD, Hydro One, TVA, Jackson EMC, EPB, Duke Energy, and Ameren.

EPRI has produced numerous tools through base programs, supplemental projects, and government projects. The DER Integration Toolkit pulls these tools together into a repository of test tools and implementation resources for applying open communication protocols to both demand response and distributed energy resources applications. EPRI continues to maintain the tools in the toolkit and provide support to members of the Information and Communication Technology for Distributed Energy Resources and Demand Response program (P161D). The goal of the toolkit is to help support development and testability of open protocols so EPRI's support of these tools extends to vendors or other stakeholders involved in member projects. The end goal is to create a "demonstration in a box" or "in-the-loop" testing where any component of the communication architecture can be simulated or implemented using components of EPRI's DER Integration Toolkit[2].

---

[1] Mounting Importance of Communications to Monitor and Control Distributed Energy Resources. EPRI, Palo Alto, CA: 2018. 3002013480.

[2] EPRI's DER Integration Toolkit: An Overview of EPRI Tools for Testing and Implementing Open Protocols. EPRI, Palo Alto, CA: 2018. 3002013623.

As grid codes and utility programs are increasingly requiring end-use devices and their control systems to use open standards it is more and more important that validation tools, simulation tools, and reference implementations are available to foster growth in the industry. Experience has shown that uptake can be slow without requirements to use a specific standard. For adoption to succeed a three-faced approach can be used.

1. Continuous technology transfer will help inform the industry of standards and teach them about the benefits of applying them.
2. Creation of open source code that vendors/manufacturers can implement in their products will likely accelerate adoption of the protocol by minimizing the time required for products to appear on the market.
3. Certification frameworks are key to ensuring protocols meet the standards.

In this project Electricité de France R&D and the Electric Power Research Institute explore the first two. The objective of this project is to develop open source code to implement IEC-61850 in both servers and clients and perform the necessary technology transfer to bring awareness of these resources. This report provides an overview of open source software to support testing and development of IEC-61850 in smart inverters. The software is still in development. This report serves as a guide for those involved in beta-testing of the software. It is intended that stakeholders and software support personnel can read this document and coordinate their efforts in the installation/deployment of the application. The purpose of this document is to:

- Describe the steps necessary to install the software.
- Understand the architecture of the different modules for both server and client.
- Describe how to use the IEC 61850 Protocol Driver Agent with systems leveraging EPRI's OpenDERMS protocol API.

The software being developed by Electricité de France R&D is an open source software stack for IEC 61850 compatible with the OpenDERMS protocol API. The purpose of the open source code is to provide developers of smart inverters, headend, control systems, or test tools with the code needed to implement IEC-61850-7-420 in their products. It will include the underlying IEC 61850 protocol but also implemented IEC-61850 information models for smart inverters (including solar and storage). The code will have well documented interfaces for developers to connect to for implementing IEC-61850 for smart inverters.

Definitions, Acronyms and Abbreviations

IEC – International Electrotechnical Commission

Qwt – The Qwt library contains GUI Components and utility classes which are primarily useful for programs with a technical background. Beside a framework for 2D plots it provides scales, sliders, dials, compasses, thermometers, wheels and knobs to control or display values, arrays, or ranges of type double.

PDA – Protocol Driver Agent

Architecture Overview

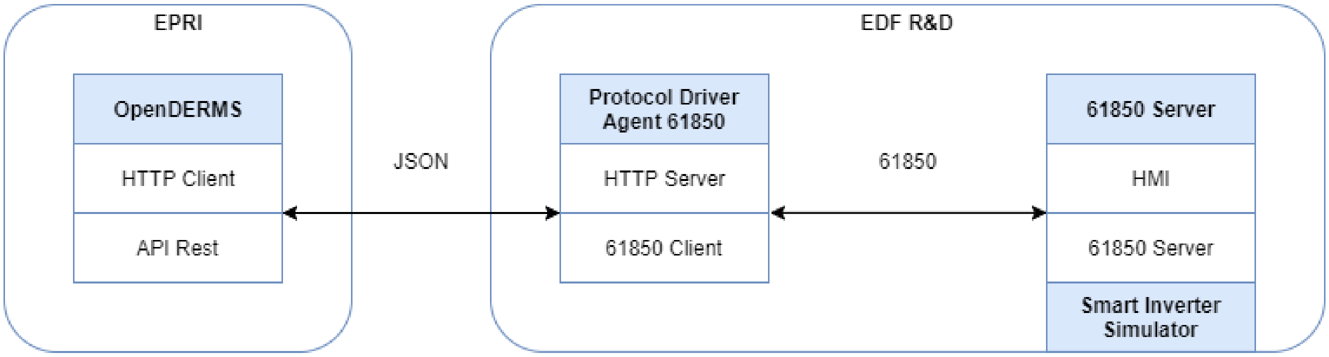

Figure 1-1
Project Architecture

The protocol driver agent following these criteria:

- Tools using the OpenDERMS API (or Boomerang add-on) sends JSON messages through HTTP protocol to a 61850 PDA which receive the requests.
- The 61850 PDA get all request parameters and parse them into 61850 requests.
- Those 61850 requests set parameters in the 61850 server data model.
- If a function is enabled, the 61850 server calls the smart inverter (or Smart Inverter Simulator) functions that generate values depending the data model.

2
INSTALLING THE SOFTWARE

Prerequisites

1. Install framework Qt: https://www.qt.io/download
2. Download source code directory (named EPRI) which contain both server and client side in C:/

Client Side Installation

1. Open Qt Creator
2. Click on "Open Project" and select pro file at:
3. C:\EPRI\OpenDerms\Epri61850 \Epri61850.pro (cf: Open an existing project)
4. Run qmake (cf: Using qmake)
5. Run application (cf: Run, compile or debug current project)
6. See how to use the application here: Client Side Overview

Server Side Installation

1. Download latest version of Qwt on: http://qwt.sourceforge.net/qwtinstall.html

   ***Note**: The default Download button on the sourceforge website doesn't link to the latest version! Version used in the project: Qwt-6.1.3*

   a. Unzip sources in C:/Qwt
   b. Open MinGW (named: Qt x.x.x for Desktop (MinGW))
   c. Move in Qwt directory: C:/Qwt
   d. Compile library
      -qmake qwt.pro
      -mingw32-make
      -mingw32-make install
2. Open Qt Creator
3. Click on "Open Project" and select pro file at:

   C:\EPRI\OpenDerms\Server61850_EPRI\Server61850_EPRI.pro (cf: Open an existing project)
4. Run qmake (cf: Using qmake)
5. Run application (cf: Run, compile or debug current project)
6. See how to use the application here: Server Side Overview

3

CLIENT SIDE OVERVIEW

This section describes the interface of the Protocol Driver Agent 61850 which receives the HTTP request and parses them in 61850 requests to the server.

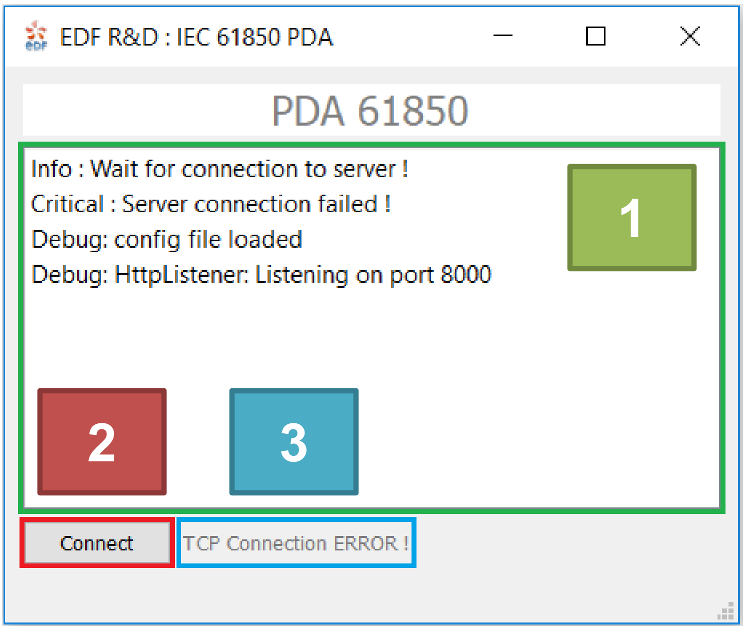

Figure 3-1
Client Side Overview

1 The window displays the application debug and information messages.

2 The “Connect” button is used to connect or disconnect the PDA from the server through the TCP connection.

3 The label displays the TCP connection state.

4
SERVER SIDE OVERVIEW

This section describes the interface of an example implementation using a test simulator which emulate a real smart inverter. This is designed to emulate EPRI's DER Inverter Simulator[3].

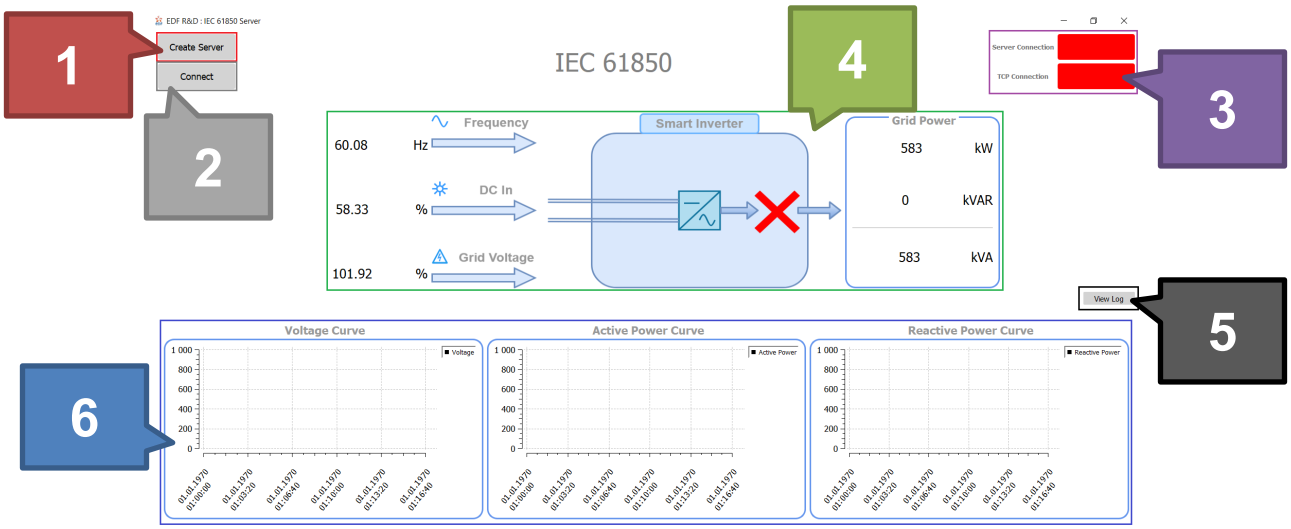

Figure 4-1
Server Side Overview

1 The "Create Server" button creates/closes the 61850 server and start the Smart Inverter Controller.

2 The "Connect" button connect/disconnect the Smart Invertor to the Grid.

3 The "Server Connection" and "TCP Connection" indicators show the state of the 61850 server and the state of the TCP connection from 61850 server to the PDA.

4 The central part displays inputs and outputs of the Smart Inverter and if it is connected to the grid.

5 The "view log" button displays the application debug and information messages.

6 Different curves to display voltage, active power and reactive power.

[3] *Overview of EPRI's DER Simulation Tool for Emulating Smart Solar Inverters and Energy Storage Systems on Communication Networks: An Overview of EPRI's Distributed Energy Resource Simulator*. EPRI, Palo Alto, CA: 2018. 3002013622.

5
USING QT

The project has been developed in C++ through the Qt framework.

Here is an overview of the different functions needed to use the application.

Open an existing project

Select 'Open Project' on the Welcome page and find the path to the .pro file of the project.

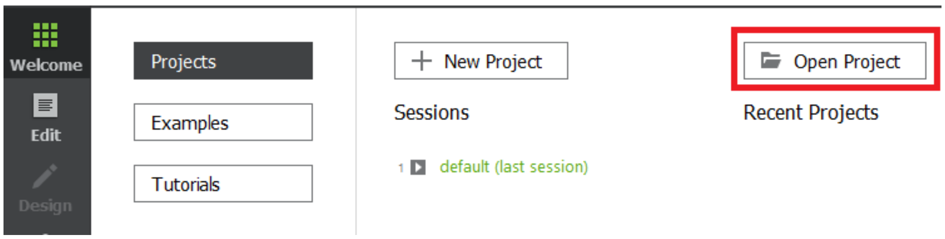

Figure 5-1
Opening an existing project

Run, compile or debug current project

On the left side of the Qt window, there is on the top all files used in the current project, and on the bottom, you can run, debug or compile the project.

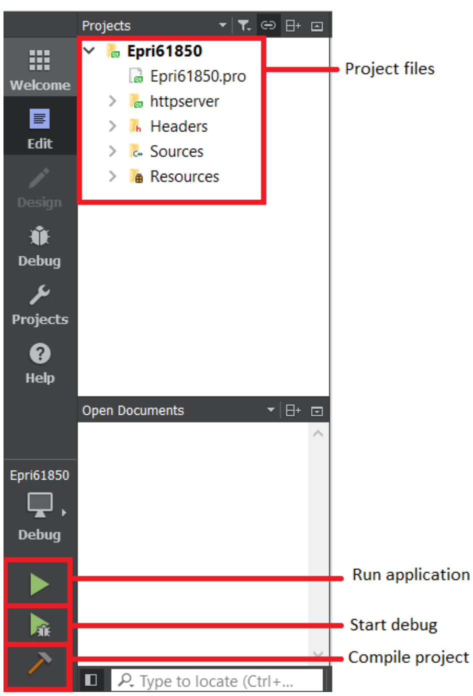

Figure 5-2
Run, compile or debug current project

Using qmake

Right-click on the main project directory on Qt and select ‘run qmake’.

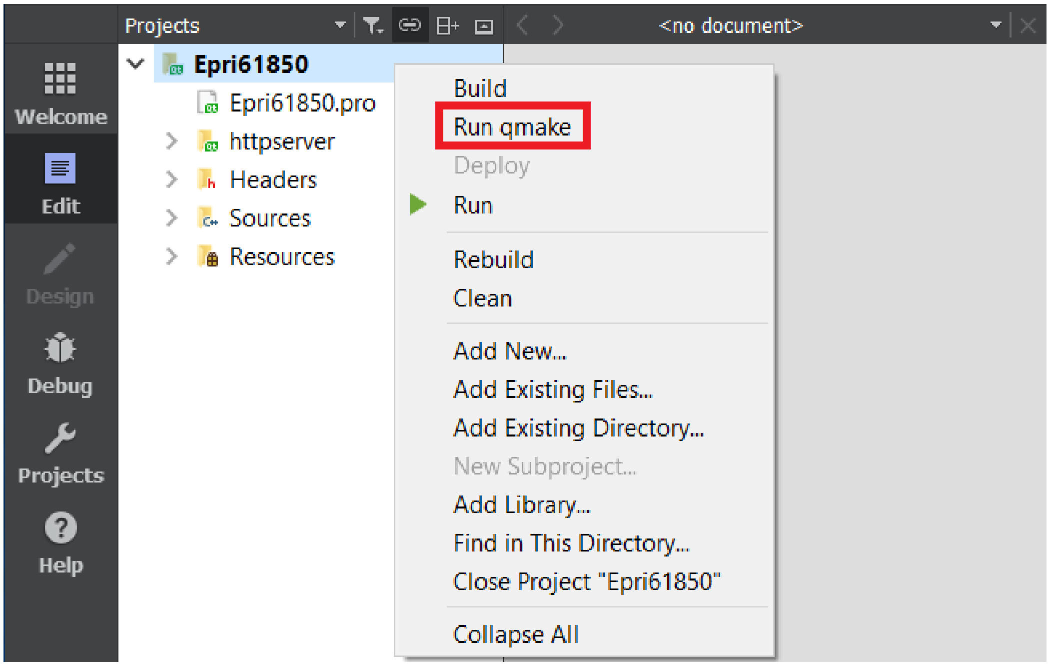

Figure 5-3
Running qmake

6
TESTING THE INSTALLATION

1. Open Qt
2. Open Server61850_EPRI project
3. Run the application
4. Create server and connect it to the Grid by clicking in “CreateServer” and “Connect” buttons
5. Open another instance of Qt
6. Open Epri61850 project
7. Run the application, the data model of the server should be displayed, if there is TCP Connection error, click on “Connect” button. If the problem remains close Qt and go back to point 5.
8. Open Google Chrome
9. Open Boomerang add-on
10. Select “configurePowerFactor” function
11. Send the request to ‘http://localhost:8000’
12. PDA Window should display the HTTP request
13. On server interface, click on “View log button”, the 61850 reference and its new value should be displayed
14. On Boomerang, select powerFactor function
15. On the request, set ‘enable’ parameter to ‘true’
16. Send the request to ‘http://localhost:8000’
17. PDA window should display all new value from Smart Inverter function
18. On server interface, a curve should be updated. If not, right click on the canvas of the curve and select “recenter”
19. On Boomerang, set ‘enable’ parameter to ‘false’
20. Send the request to ‘http://localhost:8000’ to stop the function
21. Disconnect and close PDA
22. Close server log window, stop the server and close it

7
SOFTWARE ARCHITECTURE

Below is an example of how the software can be used to integrate with other software products through either the OpenDERMS Protocol API (client) or internal memory sharing (server).

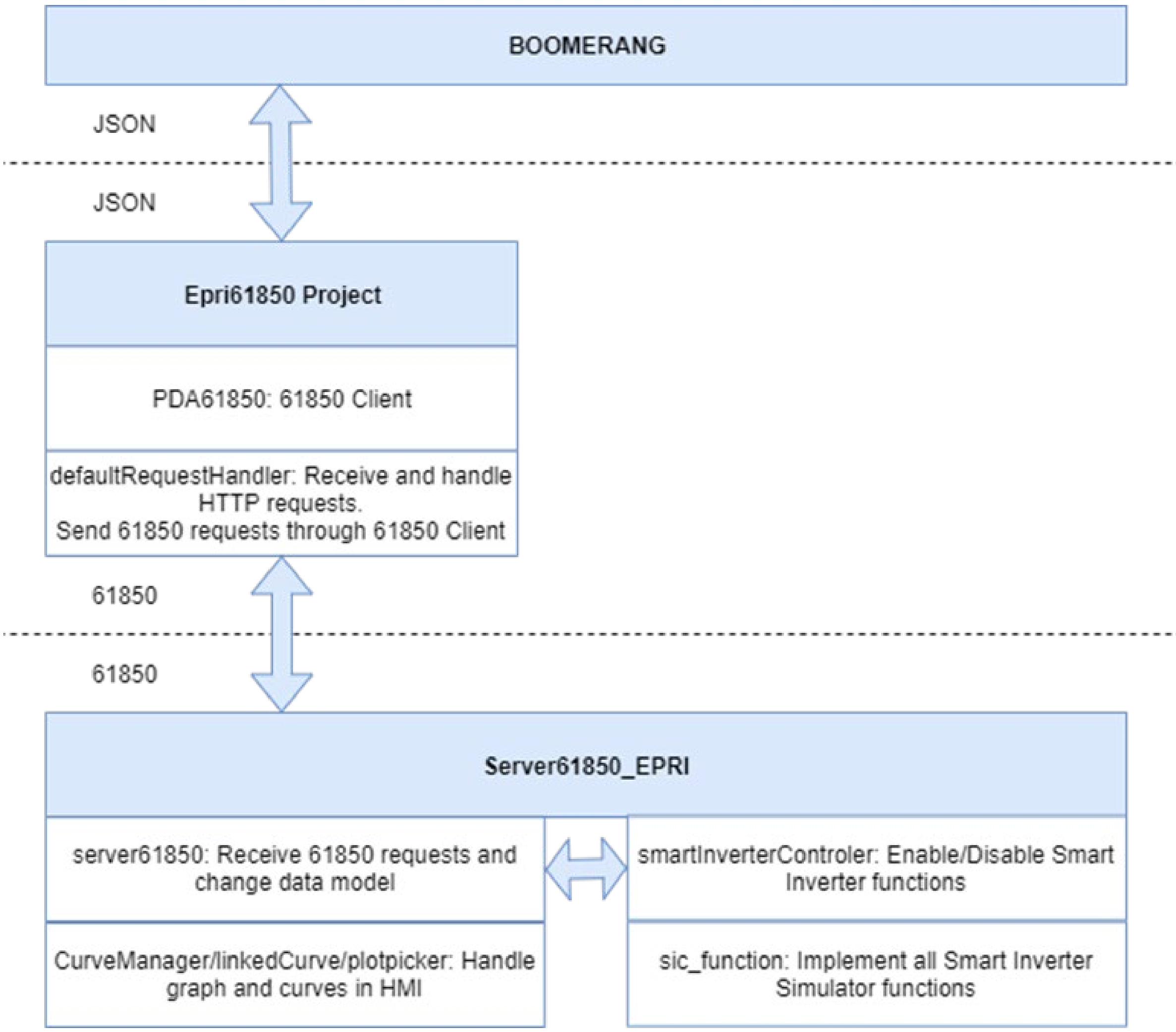

Figure 7-1
Example Software Architecture

Classes in Client Side and Server Side

Table 7-1
Client Side Software Architecture

Class name	Description
defaultRequestHandler	Inherit from HttpRequestHandler. Triggered when receive a HTTP request, read and check all JSON parameters from message and send the corresponding 61850 request to the server depending the 61850-7-420 mapping.
PDA61850	61850 Client which implement read/write/operate services and handle the TCP connection with the server.
messageLogger	Handle all log messages and writes them on the HMI.
boolean	Object Boolean which emit signal valueChanged when modify.

Table 7-2
Server Side Software Architecture

Class name	Description
Server61850	Create 61850 server through .icd configuration file. Receive 61850 requests from client and write data model. Create the HMI.
messageLogger	Handle all log messages and writes them on the HMI.
messageLoggerWindow	Handle all log messages and writes them on a detached window.
boolean	Object Boolean which emit signal valueChanged when modify.
curvesManager	Create graph for curves in the HMI.
linkedCurve	Link curves to 61850 reference. If the 61850 value is updated, the curve will be updated as well.
plotPicker	Class to display a label to display x and y value where the mouse is pointing on a graph.
timeScaleDraw	class used to set time format in x-axis on a graph.
smartInveterControler	Enable/Disable Smart Inverter Simulator functions *(temporary before translating inverter.css in python)*
sic_function	Implement all Smart Inverter Simulator functions *(temporary before translating inverter.css in python)*

8

IEC 61850-7-420 DATA MODEL

In this table, you will find all OpenDERMS functions and their parameters associated with their 61850-7-420 Logical nodes and Data Object.

OpenDERMS Functions	Functions Parameters	Logical Node	Data Object
startCommunication	Only for TCP Connection	/	/
stopCommunication	Only for TCP Connection	/	/
configureConnectDisconnect	reversionTimeout	/	/
	timeWindow	/	/
connect		CSWI	pos
disconnect		CSWI	pos
configureAdjustMaxGen	timeWindow	DWGC	WinTms
	reversionTimeout	DWGC	RvrtTms
	rampTime	DWGC	RmpTms
	maxOutput	DWGC	GnWSpt
configurePowerFactor	timeWindow	DFPF	WinTms
	reversionTimeout	DFPF	RvrtTms
	rampTime	DFPF	RmpTms
	powerFactor	DFPF	DftPFTgt
	varAction	DFPF	X
configureChargeDischarge	timeWindow	DTCD	WinTms
	reversionTimeout	DTCD	RvrtTms
	rampTime	DTCD	RmpTms
	chargeDischargeRate	DTCD	RmpUpRte
	chargeDischargeRate	DTCD	RmpDnRte
	capacityRating	DTCD	WhRtg
	storageReserve	DTCD	X
	maxStateOfCharge	DTCD	SocUseMaxPct
	minStorageReserve	DTCD	UseCapMinPct
	minBatteryChargeVoltage	DTCD	X
	maxApparentChargingPower	DTCD	X
	activePowerChargeGradient	DTCD	X
configurePriceSignal	timeWindow	DPRG	WinTms
	reversionTimeout	DPRG	RvrtTms
	rampTime	DPRG	RmpTms
	priceSignal	DPRG	X
configureStorageSettings	maxActivePower	DLOD	Wmax
	maxApparentPower	DLOD	VAMax
	maxReactivePower	DLOD	VArMax
	maxRamp	DLOD	X
	minStorageReserve	DLOD	X
	maxChargeRate	DLOD	RampUpMaxLodRtg
	maxDischargeRate	DLOD	RampDnMaxLodRtg
	maxApparentChargingPower	DLOD	VAMaxRtg
	maxStateOfCharge	DLOD	SocUseMaxPct
configureConstantVarModes	timeWindow	DVAR	WinTms
	reversionTimeout	DVAR	RvrtTms
	rampTime	DVAR	RmpTms

	referenceReactivePower	DVAR	VArSetRef
	setPoint	DVAR	X
	percentOfMaxVARS	DVAR	VArTgtPct
	percentOfAvailableVARS	DVAR	X
configureDynamicReactiveCurrentMode	gradientMode	DRGS	ArGraMod
	deadbandMinVoltage	DRGS	DbVMin
	deadbandMaxVoltage	DRGS	DbVMax
	gradientForSags	DRGS	ArGraSag
	gradientForSwells	DRGS	ArGraSwl
	filterTime	DRGS	FilTms
	eventBasedActiveCurrentSupport	DRGS	X
	holdTime	DRGS	HoldTmms
	blockZoneVoltage	DRGS	BlkZnV
configureRealPowerSmoothing	timeWindow	DWSM	WinTms
	reversionTimeout	DWSM	RvrtTms
	rampTime	DWSM	RmpTms
	gradient	DWSM	X
	filterTime	DWSM	FilTms
	lowerLimit	DWSM	X
	upperLimit	DWSM	X
configureDynamicVoltWattMode	reversionTimeout	DVWC	RvrtTms
	referenceVoltage	DVWC	Vref
	referenceVoltageOffset	DVWC	VRefOfs
	maxActivePower	DVWC	Wmax
	filterTime	DVWC	FilTms
	lowerDeadband	DVWC	X
	upperDeadband	DVWC	X
configurePeakPowerLimiting	timeWindow	DPKP	WinTms
	reversionTimeout	DPKP	RvrtTms
	rampTime	DPKP	RmpTms
	peakPowerLimit	DPKP	X
configureLoadGenerationFollowing	timeWindow	DLFL	WinTms
	reversionTimeout	DLFL	RvrtTms
	rampTime	DLFL	RmpTms
	followingRatio	DLFL	X
configureGenericCurves	curveIndexSelector	DGSM	X
	curveId	DGSM	InCrv
	numberOfPoints	DGSM	X
	timeWindow	DGSM	WinTms
	reversionTimeout	DGSM	RvrtTms
	rampTime	DGSM	RmpTms
	modeType	DGSM	ModTyp
	independentUnits	DGSM	IndpUnt
	dependentUnits	DGSM	DepRef
	timeConstant	DGSM	x
	decreasingMaxRampRate	DGSM	RmpDecTmm
	increasingMaxRampRate	DGSM	RmpIncrTmm
	releaseMaxRampRate	DGSM	RmpRUp
	dependentVariableSnapshotStart	DGSM	DepRefStr

	dependentVariableSnapshotStop	DGSM	DepRefStop
	mrcSagSwell	DGSM	X
	xValue1	DGSM	X
	yValue1	DGSM	X
configureVoltWattCurve	Points[volt, watt]	DVWC	VWCrv
configureVoltVarCurve	Points[volt, var]	DVVR	VVArCrv
configureFreqWattCurve	Points[freq, watt]	DLFW	HzWCrv

X = Not found in IEC 61850-7-420
/ = Don't affect IEC 61850-7-420
Ref = May not be in the LN

9 TROUBLESHOOTING

You should find log files at:

- C:\EPRI\OpenDerms\Epri61850\Log.txt
- C:\EPRI\OpenDerms\Server61850_EPRI\Log.txt

A log message looks like:

[Date of message] Debug level: Message [File concerned (function concerned): line concerned]

For example:

[10/31/2018 16:06:17] Debug: DO: EventLog [..\Epri61850\epri61850.cpp (void Epri61850: Connect ()): line 35

10
ONGOING DEVELOPMENT

The industry continues to improve current communications standards to meet modern use cases for solar, energy storage, and demand response systems. As the industry evolves EPRI plans to adapt the DER Integration Toolkit, including the Distributed Energy Resources Simulator, to meet the needs of the industry.

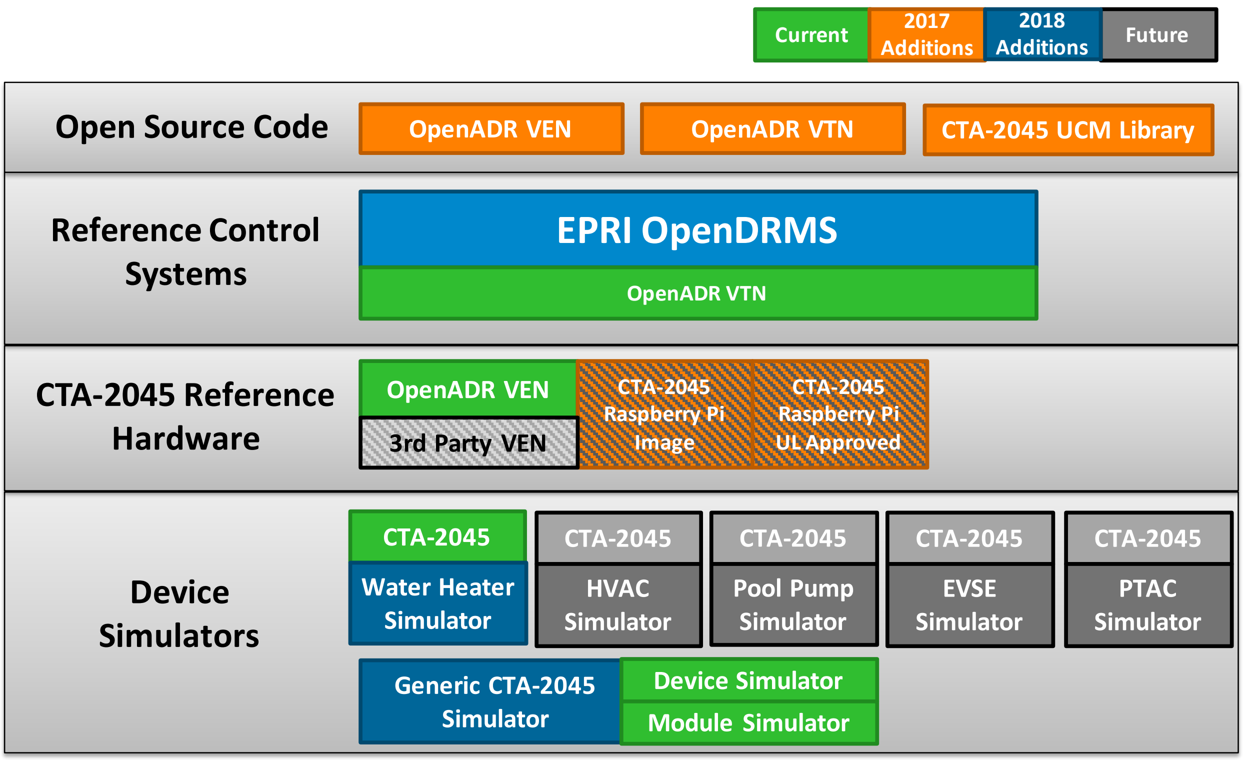

Figure 10-1
Upcoming tools for demand response technologies.

In the immediate term EPRI plans to continue to expand the device simulators to support additional device types including energy storage, heating and cooling systems, pool pumps, electric vehicles, and commercial room air conditioners. EPRI is also looking to expand protocol support across the entire toolkit to include all relevant communications protocols. The reference control systems and device simulators have been intentionally designed to scale well to include new protocols as needs arise. EPRI is funded through base research programs, supplemental projects with utilities, and government funded activities to expand the tools in the DER toolkit.

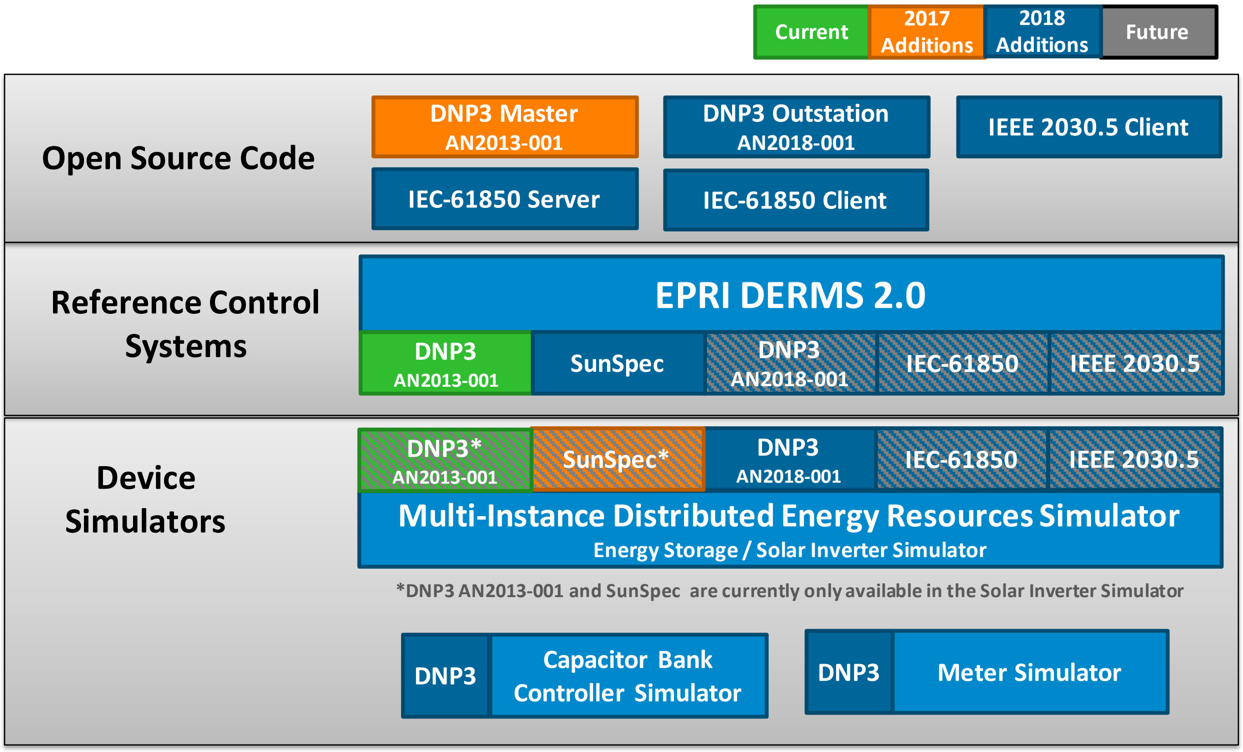

Figure 10-2
Upcoming tools for inverter-based technologies.

In addition to adding new tools EPRI is constantly receiving feedback from users of the toolkit including feature requests and bugs. This feedback is reviewed and included in the work plan for each tool. Severe bugs are addressed immediately.

EPRI is working with Sandia National Laboratory in designing a distributed monitoring system that can patrol a wide range of cyber-attack vectors, detect various attack methods, predict adversary movements, and implement controls that mitigate damage to DER devices. As part of this project, EPRI will update the DER Smart functions to latest IEEE 1547 functions. EPRI will also upgrade the DNP3 outstation to implement AN2018-001 standard and will add Transport Layer Security (TLS) to the Modbus protocol driver. EPRI is also working with EDF Energy on integrating IEC 61850 drivers to OpenDERMS and the device simulators in the next year. Another development work planned for 2019 is an IEEE 2030.5 to SunSpec Modbus translator. To meet the recent requirements of Rule – 21 and IEEE 1547, there is much interest among DER vendors to develop an IEEE 2030.5 to Modbus translator. EPRI will extend the capability of the currently available, open source IEEE 2030.5 client, to a translator that can communicate with SunSpec Modbus compliant devices. As part of this project, a specification that documents the mapping between the two communication protocols will also be developed.

EPRI is committed to the development of test tools to ease the entry of open protocols into the market and plans to continue to develop, maintain, and support these tools for members. The end goal is to create a “demonstration in a box” where any component of the communication architecture can be validated, tested, or implemented using components of the DER Integration Toolkit. More information on other tools in EPRI’s DER Integration Toolkit can be found in the toolkit summary report[2].

Export Control Restrictions

Access to and use of this EPRI product is granted with the specific understanding and requirement that responsibility for ensuring full compliance with all applicable U.S. and foreign export laws and regulations is being undertaken by you and your company. This includes an obligation to ensure that any individual receiving access hereunder who is not a U.S. citizen or U.S. permanent resident is permitted access under applicable U.S. and foreign export laws and regulations.

In the event you are uncertain whether you or your company may lawfully obtain access to this EPRI product, you acknowledge that it is your obligation to consult with your company's legal counsel to determine whether this access is lawful. Although EPRI may make available on a case by case basis an informal assessment of the applicable U.S. export classification for specific EPRI products, you and your company acknowledge that this assessment is solely for informational purposes and not for reliance purposes.

Your obligations regarding U.S. export control requirements apply during and after you and your company's engagement with EPRI. To be clear, the obligations continue after your retirement or other departure from your company, and include any knowledge retained after gaining access to EPRI products.

You and your company understand and acknowledge your obligations to make a prompt report to EPRI and the appropriate authorities regarding any access to or use of this EPRI product hereunder that may be in violation of applicable U.S. or foreign export laws or regulations.

The Electric Power Research Institute, Inc. (EPRI, www.epri.com) conducts research and development relating to the generation, delivery and use of electricity for the benefit of the public. An independent, nonprofit organization, EPRI brings together its scientists and engineers as well as experts from academia and industry to help address challenges in electricity, including reliability, efficiency, affordability, health, safety and the environment. EPRI members represent 90% of the electric utility revenue in the United States with international participation in 35 countries. EPRI's principal offices and laboratories are located in Palo Alto, Calif.; Charlotte, N.C.; Knoxville, Tenn.; and Lenox, Mass.

Together…Shaping the Future of Electricity

3002013625

Electric Power Research Institute
3420 Hillview Avenue, Palo Alto, California 94304-1338 • PO Box 10412, Palo Alto, California 94303-0813 • USA
800.313.3774 • 650.855.2121 • askepri@epri.com • www.epri.com